

TITLE OF THE INVENTION

**PRINT JOB MANAGEMENT APPARATUS AND PRINT JOB MANAGEMENT
METHOD**

5 BACKGROUND OF THE INVENTION**1. Field of the Invention**

The present invention relates to a print job management apparatus and a print job management method that manage print jobs which are executed by a printing device.

10 2. Description of the Prior Art

A print server for storing print jobs printed by a printer has been proposed as a print job management apparatus (see, for example, Japanese Patent Laid-Open Gazette No. 09-269879). This print server enables each print job stored therein to be
15 reprinted by the printer, in response to a user's reprinting request.

The prior art print server requires a high-capacity storage device for storing print jobs of high-volume image data, such as photographic images. In such cases, the number of print
20 jobs stored in the print server is restricted. This problem is found not only in the case of storage of print jobs for reprinting but in the case of storing newly received print jobs of high-volume image data and in the case of receiving storing

a large number of print jobs simultaneously.

SUMMARY OF THE INVENTION

The object of the present invention is to provide a print
5 job management apparatus and a corresponding method that save
the storage capacity of print jobs to solve the problem of the
prior art mentioned above. The object of the present invention
is further to provide a print job management apparatus and a
corresponding method that efficiently manage print jobs.

10 In order to achieve at least part of aforementioned
objects, the present invention is structured as follows.

A print job management apparatus of the present invention
is an apparatus that manages print jobs, which are executed
by a printing device, the print job management apparatus
15 including: a job acceptance module that receives each print
job with image data; a job storage module that has a capacity
of storing multiple print jobs received by the job acceptance
module; a redundant data retrieval module that retrieves
redundant image data among image data of print jobs stored in
20 the job storage module; and a job status setting module that
leaves at least one of the redundant image data retrieved by
the redundant data retrieval module while deleting the other
of the retrieved redundant image data, and sets a status of

a print job in which the image data was deleted, to be executable with remaining image data by the printing device.

The print job management apparatus of the invention receives each print job with image data and stores the received
5 print job into the job storage module, which has the capacity of storing multiple print jobs. The print job management apparatus retrieves redundant image data among image data of print jobs stored in the job storage module, and leaves at least one of the retrieved redundant image data while deleting the
10 other of the retrieved redundant image data. The print job management apparatus sets the status of a print job in which the image data was deleted, to be executable with remaining image data by the printing device. The technique of the invention deletes redundant image data while ensuring
15 execution of a print job in which the image data was deleted. This arrangement effectively saves the storage capacity of print jobs.

In one preferable application of the job management apparatus of the invention, each print job includes
20 identification information for identifying the image data, and the redundant data retrieval module retrieves image data having identical identification information among the image data of the print jobs stored in the job storage module, as the

redundant image data. Here, the identification information may include at least one of a file name of each image data, size of image data, identification information for identifying a digital camera used to record the image data, and date of
5 recording the image data with the digital camera.

In another preferable application of the print job management apparatus of the invention, the job status setting module sets the status of the print job in which the image data was deleted, such that the remaining image data among the
10 redundant image data is shared by a print job having the remaining image data and the print job in which the image data was deleted. In this application, each print job may be executable by the printing device by utilizing reference data, which is generated for reference to image data of the print
15 job in the process of storage into the job storage module, to read the image data stored in the job storage module, and the job status setting module may overwrite reference data of the print job in which the image data was deleted, with reference data of a print job having the remaining image data among the
20 redundant image data.

In still another preferable applications of the print job management apparatus of the invention, the job status setting module preferentially deletes image data stored

earlier in the job storage module, or preferentially deletes printed image data, among the redundant image data.

In one preferable embodiment, the print job management apparatus of the invention further includes an image processing
5 module that makes image data of each print job, which is stored in the job storage module, subjected to a preset series of image processing and thereby converts the image data into print data printable by the printing device, and the redundant data retrieval module and the job status setting module respectively
10 execute the retrieval and the deletion and setting the status of the print job, while the image processing module is not activated.

In another preferable embodiment, the print job management apparatus of the invention further includes a job
15 deletion module that deletes a print job stored in the job storage module at a preset timing. In this embodiment, the preset timing may be any of a timing when a total number of print jobs stored in the job storage module reaches a preset level, a timing when a total storage capacity of print jobs
20 stored in the job storage module reaches a preset volume, and a timing when a duration of storage of each print job stored in the job storage module reaches a preset time period. As one application of this embodiment, the job deletion module

may preferentially delete a print job stored earlier, among the print jobs stored in the job storage module.

The technique of the present invention is not restricted to the print job management apparatus described above, but is
5 also applicable to a print job management method.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 schematically illustrates the configuration of a photo studio system 10; Fig. 2 shows an example of print
10 management data; Fig. 3 is a flowchart showing a print job acceptance routine; Fig. 4 shows print jobs stored in the job folder 28a; Fig. 5 shows an example of a job execution status display window 90; Fig. 6 is a flowchart showing a print job management routine; Fig. 7 shows deletion of redundant image
15 data; and Fig. 8 is a flowchart showing a print job deletion routine.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

One mode of carrying out the invention is discussed below
20 as a preferred embodiment. Fig. 1 schematically illustrates the configuration of a photo studio system 10 that includes a print server 20 functioning as a print job management apparatus in one embodiment of the invention. The photo studio

system 10 includes two printers 40 and 50, the print server 20 of the embodiment that connects with the two printers 40 and 50, and multiple client computers 60 that output print jobs of image data to the print server 20. The multiple client
5 computers 60 are connected with the print server 20 via a LAN 12. The photo studio systems 10 are respectively located in multiple regions and are under management of a management server 70 connected via the Internet 80.

Each of the client computers 60 is constructed as a
10 general computer. The client computer 60 is connected with a digital still camera 62 via, for example, a USB interface cable, to receive image data recorded with the digital still camera 62, and stores the input image data into a storage medium like a hard disk of the client computer 60.

15 The print server 20 of the embodiment is constructed as a general computer including a CPU 21. The CPU 21 is connected via a bus 30 with a ROM 22 that stores various processing programs, a RAM 23 that temporarily stores data, a clock generator 24 that generates an operation clock, a video board
20 25 that has a connector to a display 26, a LAN board 27 that has a connector to the LAN 12, and an input-output controller 29 that is linked with a hard disk 28 and the printers 40 and 50. The LAN board 27 is connected with a router 14 via the

LAN 12. The print server 20 of the embodiment gains access to the Internet 80 via the router 14.

The hard disk 28 of the print server 20 in this embodiment has a job folder 28a that stores print jobs sent from the client
5 computer 60, and spoolers 28b and 28c that respectively spool at most two print data or print jobs, which are printable with either of the printers 40 and 50 and include a currently executing print job. The hard disk 28 also has a status folder 28d that stores status data regarding the sizes and types of
10 papers currently set in the printers 40 and 50, a state of paper jam, and the numbers of print jobs currently spooled in the spoolers 28b and 28c, and a program folder 28e that stores various processing programs.

Each of the print jobs stored in the job folder 28a of
15 the hard disk 28 has an image file including object image data to be printed, a print management file including description of print management data regarding specifications of a paper size, a paper type, an image layout, and a printer used for printing, and an ICC profile. The image file has an image area
20 for storing image data and an attribute information area for storing attribute information relating to the image data. The attribute information stored in the attribute information area includes, for example, the size of image data, identification

information for identifying the digital still camera 62 used for taking the image data, and the date of taking the image data. In response to reception of a print job from any of the client computers 60, the print server 20 of the embodiment reads
5 the attribute information from the attribute information area of the image file in the received print job and writes the attribute information as image identification data into the print management file. Fig. 2 shows an example of the print management data written in the print management file. As shown
10 in Fig. 2, the print management data includes a job ID uniquely allocated to each print job, the date of job acceptance, a reference path, specifications of a paper size, a paper type, an image layout, and a printer, an execution status of the print job, and image identification data. The image identification
15 data includes the name of an image file, the size of image data, a camera ID, and the date of photographing.

The processing programs stored in the program folder 28e of the hard disk 28 include printer drivers executed to control the printers 40 and 50 respectively, a color converter executed
20 to generate a color correction table based on an ICC profile of a selected printer for printing and an ICC profile of object image data and to carry out color correction of the object image data with the generated color correction table, and a Web

browser executed to browse various pieces of information.

The description now regards the operations of the photo studio system 10 constructed as discussed above, especially the operations of the print server 20 of the embodiment. A print job acceptance routine shown in the flowchart of Fig. 3, a print job management routine shown in the flowchart of Fig. 6, and a print job deletion routine shown in the flowchart of Fig. 8 are successively described as the operations of the print server 20 of the embodiment.

10 The print job acceptance routine is activated, in response to reception of a signal representing transmission of a print job from one of the client computers 60. When the print job acceptance routine shown in the flowchart of Fig. 3 starts, the CPU 21 of the print server 20 of the embodiment
15 receives a print job sent from the client computer 60 (step S100), allocates a job ID to the received print job (step S102), and stores the print job with the allocated job ID into the job folder 28a of the hard disk 28 (step S104). The print job acceptance routine is then terminated. The job ID is allocated
20 to each print job in the order of reception and is written into the print management file. Fig. 4 shows image files and print management files of print jobs stored in the job folder 28a. In the illustrated example of Fig. 4, ID folders identified

by job IDs as folder names are created in a directory 'DATA', which is a lower hierarchy of a directory 'JOB' in the job folder 28a. Each image file is stored in the corresponding ID folder. Print management files identified by job IDs as file names are
5 stored in a directory 'MANAGE', which is also the lower hierarchy of the directory 'JOB' in the job folder 28a. The print job execution procedure of this embodiment reads object image data by referring to a reference path (shown in brackets in Fig. 4) written in a corresponding print management file,
10 and converts the object image data into print data printable with a selected one of the printers 40 and 50.

When each received print job is stored in the job folder 28a according to the print job acceptance process discussed above, it is determined whether the paper size and the paper
15 type specified in the print job are consistent with the paper size and the paper type currently set in either of the printers 40 and 50. In the case of consistency, the execution status of the print job is set to a print queuing state. In the case of inconsistency, on the other hand, the execution status of
20 the print job is set to a pause state. When the paper size and the paper type set in either of the printers 40 and 50 are changed to be consistent with the paper size and the paper type specified in the print job, the execution status of the print

job is changed from the pause state to the print queuing state. When the print job set in the print queuing state is present in the job folder 28a, it is determined whether either of the spoolers 28b and 28c corresponding to the printers 40 and 50 has any vacancy for spooling new print data. The print job in the print queuing state is allocated to a selected printer corresponding to the spooler having any vacancy for spooling new print data. The execution status of the print job is then changed from the print queuing state to an allocated state.

10 In response to allocation of the print job to the selected printer, the procedure generates a color correction table based on the ICC profile of the selected printer and the ICC profile of the print job, reads object image data to be printed by referring to the reference path of the print job, and carries

15 out color correction of the object image data with the generated color correction table. The procedure activates the printer driver corresponding to the selected printer to make the color-converted image data subjected to color conversion into CMYK data and a halftoning process and converted into print

20 data printable with the selected printer. The converted print data is transferred as a standby print job to the spooler corresponding to the selected printer. On completion of a currently executing print job, each of the spoolers 28b and

28c outputs print data spooled therein as a standby print job to the corresponding one of the printers 40 and 50 to start printing. The execution status of the print job is then changed from the allocated state to a printing state. On completion
5 of execution of the print job, the execution status of the completed print job is changed from the printing state to a printed state. The execution status of the print job is written into the print management file of the print job stored in the job folder 28a. Even after completion of printing with either
10 of the printers 40 and 50, the print job in the printed state is kept in the job folder 28a for reprinting.

In response to a requirement from the print server 20 or each of the client computers 60, data regarding the execution statuses of respective print jobs are read from the
15 corresponding print management files stored in the job folder 28a and are provided in the form of a Web page. Fig. 5 shows a job execution status display window 90 opening on a display of the print server 20 or on a display of each client computer 60. The job execution status display window 90 includes a job
20 ID display column 91 showing job IDs of respective print jobs, an execution status display column 92 showing execution statuses of the respective print jobs, a reference path display column 93 showing reference paths of object image data to be

printed, a paper specifications display column 94 showing specifications of paper size and paper type in the respective print jobs, a printer name display column 95 showing names of specified printers, a job acceptance date display column 96
5 showing dates of acceptance of the respective print jobs, and an update button 97 to effectuate new entries regarding, for example, the execution statuses and the paper specifications of the print jobs. For the better understanding, different symbols of 'Printing' state, 'Standby' state, 'Print Queuing' state, 'Pause' state, and 'Printed' state are set corresponding
10 to the 'printing state', the 'allocated state', the 'print queuing state', the 'pause state', and the 'printed state' discussed above and are shown in the execution status display column 92 of the respective print jobs on the job execution
15 status display window 90 of Fig. 5. In the actual display, however, these states are distinctly shown in color lighting: for example, blinking green circle for the 'Printing' state, lighting green circle for the 'Standby' state, lighting yellow circle for the 'Print Queuing' state, lighting orange circle
20 for the 'Pause' state, and lighting black circle for the 'Printed' state. In the illustrated example of Fig. 5, the user is allowed to open a pulldown menu and enter a selected state in the execution status display column 92. A change of

the execution status of a print job from the 'printed state' to the 'print queuing state' enables the print job in the printed state to be reprinted as the print job in the print queuing state with either of the printers 40 and 50.

5 The print job management routine is repeatedly executed while no print job is executed, that is, when no print job in the printing queuing state is present in the job folder 28a. Execution of this print job management routine only during non-execution of any print job effectively protects the print
10 server 20 from excessive processing load. When the print job management routine shown in the flowchart of Fig. 6 starts, the CPU 21 of the print server 20 of the embodiment successively specifies a latest print job having a largest job ID number as an object job and reads image identification data for
15 identifying object image data included in the object job from its corresponding print management file (step S200). The CPU 21 then retrieves any print job having identical image identification data with the image identification data of the specified object job (step S202). The image identification
20 data includes the name of an image file, the size of image data, the camera ID, and the date of photographing as shown in Fig. 2. Two image data having identical image identification data mean that the two image data were taken with an identical

digital still camera 62 on an identical date and have an identical data size and an identical file name. Namely these two image data are regarded as identical. Retrieval of any print job having identical image identification data
5 accordingly determines whether any print job having redundant image data has been received in the past.

When any print job having the identical image identification data is hit as a result of retrieval at step S204, the CPU 21 determines whether the execution status of
10 the retrieved print job is in the printed state (step S206). When the retrieved print job is set in the printed state, the CPU 21 deletes image data present on the reference path of the corresponding print management file of the retrieved print job (step S208) and overwrites the reference path of the print
15 management file of the retrieved print job with the reference path of the print management file of the object job specified at step S200 (step S210). This process leaves only image data of the object job among multiple print jobs having redundant image data, while deleting the redundant image data of the other
20 print job. The other print job with the deleted image data is executable by referring to the image data of the object job. This arrangement effectively prevents identical image data from being stored redundantly in the job folder 28a and thus

desirably saves the storage capacity of print jobs. Fig. 7 shows deletion of redundant image data. In the illustrated example of Fig. 7, image data 'dddd.tif' of a print job (object job) with job ID '1031' is identical with image data 'dddd.tif' of a print job with job ID '1019'. The process accordingly deletes the image data of the print job with the job ID '1019' and overwrites the reference path (shown in brackets in Fig. 7) of the print management file of the print job with the job ID '1019' with the reference path of the object job with the job ID '1031'. The print job with the job ID '1019' is executed by reading the image data of the print job with the job ID '1031' from the overwritten reference path. Namely the image data present on the reference path '../DATA/1031/dddd.tif' is shared by the print job with the job ID '1031' and the print job with the job ID '1019'.

The processing of steps S208 and S210 is skipped when no print job having the identical image identification data is hit as a result of retrieval at step S204 or when the retrieved print job is not in the printed state at step S206. The routine of steps S200 to S210 is repeated until the processing has been completed for all the print jobs. When the processing has been completed for all the print jobs (step S212), the CPU 21 exits from the print job management routine.

The print job deletion routine is executed repeatedly at preset time intervals (for example, at every 24 hours). When the print job deletion routine shown in the flowchart of Fig. 8 starts, the CPU 21 of the print server 20 of the embodiment
5 reads the job acceptance dates from the print management files of all the print jobs stored in the job folder 28a (step S300) and determines whether there is any print job having the job acceptance date that is a preset number of days (for example, 30 days) behind the current date (step S302). When there is
10 any print job having the job acceptance date that is the preset number of days behind, the CPU 21 determines whether the relevant print job is in the printed state (step S304). When the relevant print job is in the printed state, the CPU 21 deletes the relevant print job (step S306) and exits from the
15 print job deletion routine. A concrete procedure of deletion of a relevant print job deletes a corresponding image file and a corresponding print management file with a key of the job ID allocated to the relevant print job having the job acceptance date that is the preset number of days behind. For example,
20 it is assumed that a print job with a job ID '1020' is deleted from the state of Fig. 7. The deletion process deletes a folder '1020' corresponding to the job ID '1020' and storing image data 'aaaa.tif' and a print management file '1020.JOB'

corresponding to the job ID '1020'. In the case where the corresponding image file has already been deleted by the print job management routine of Fig. 6, the deletion process deletes only the corresponding print management file with the key of the job ID. As described previously with reference to the print job management routine of Fig. 6, one image file may be shared by a print job originally having the image file as well as another print job. The print job management routine of Fig. 6 leaves the image data of a latest print job (that is, a print job having a largest job ID number) among the redundant image data, while the print job deletion routine of Fig. 8 preferentially deletes the image data of a print job stored earlier in the job folder 28a (that is, a print job having a smaller job ID number). The process of deleting the print job with the key of the job ID accordingly does not mistakenly delete the image data shared by multiple print jobs. When there is no print job having the job acceptance date that is the preset number of days behind the current date at step S302, the CPU immediately exits from the print job deletion routine.

As described above, the print server 20 of the embodiment retrieves any print job having identical image identification data for identifying image data with the image identification data of an object job, which is successively specified among

all the print jobs stored in the job folder 28a, deletes the image data of the retrieved print job, and overwrites the reference path of the retrieved print job with the reference path of the object job. This arrangement desirably eliminates
5 redundancy of image data and saves the storage capacity of the job folder 28a. The print job with the deleted image data is executable by referring to the image data of the object job. The image identification data including the name of an image file, the size of image data, the camera ID, and the date of
10 photographing is used for retrieval of any print job having the redundant image data. This ensures retrieval of redundant image data with high accuracy.

When redundant image data are present in the job folder 28a, the print server 20 of the embodiment leaves the image
15 data of a latest print job (that is, a print job having a largest job ID number) among the redundant image data. This arrangement ensures storage of latest image data in the job folder 28a. The print server 20 also preferentially deletes an older print job (that is, a print job having a smaller job
20 ID number). This arrangement ensures adequate management of print jobs.

The CPU 21 of the print server 20 of the embodiment executing the print job acceptance routine of Fig. 3

corresponds to the job acceptance module of the invention. The job folder 28a of the hard disk 28 that stores print jobs corresponds to the job storage module of the invention. The CPU 21 of the print server 20 executing the processing of steps 5 S200 through S204 in the job management routine of Fig. 6 corresponds to the redundant data retrieval module of the invention. The CPU 21 executing the process of steps S208 and S210 in the print job management routine of Fig. 6 corresponds to the job status setting module of the invention.

10 The print server 20 of the embodiment retrieves any print job of redundant image data based on the image identification data, which includes the name of an image file, the size of image data, the camera ID, and the date of photographing. A combination of three or a less number of parameters selected 15 among the name of an image file, the size of image data, the camera ID, and the date of photographing may be used for retrieval of any print job having redundant image data, although the accuracy of retrieval is slightly lowered. A combination of any other parameters may be applied to retrieval 20 of any print job having redundant image data.

The print server 20 of the embodiment deletes the image data of only the print job in the printed state at steps S206 and S208 in the print job management routine of Fig. 6. One

possible modification may delete image data of a print job in an execution status other than the printed state, as long as the print job shares the image data of another print job for execution of printing.

5 The print server 20 of the embodiment executes the print job management routine of Fig. 6 while no print job is executed, that is, when no print job in the print queuing state is present in the job folder 28a. The print job management routine may be executed in parallel with another series of processing, if
10 the print server 20 has marginal processing capacity.

 The print server 20 of the embodiment leaves the image data of a latest print job (that is, a print job having a largest job ID number) among the redundant image data in the print job management routine of Fig. 6. This is, however, not
15 restrictive, and image data of another print job may be left instead. In such modification, the reference path of the print job with the deleted image data is overwritten with the reference path of the print job having the remaining image data. In this case, the print job deletion routine of Fig. 8 is to
20 be modified not to delete a print job having image data shared by another print job.

 The print server 20 of the embodiment deletes a print job having the job acceptance date that is a preset number of

days behind the current date among the print jobs stored in the job folder 28a in the print job deletion routine of Fig. 8. One possible modification may delete a print job when the total number of accepted print jobs reaches a preset level or
5 when the total storage capacity of print jobs reaches a preset volume. In such modification, a print job accepted earlier (that is, a print job having a smaller job ID number) may be deleted preferentially.

In the above embodiment, the technique of the present
10 invention is applied to the print server 20 that functions as a print job management apparatus. The technique of the invention is also applicable to a print job management method in relation to the print job management apparatus of the embodiment. Still another application is a program that
15 causes the computer to function as the print job management apparatus of the embodiment. In the case of such programs, the respective steps of the print job acceptance routine, the print job management routine, and the print job deletion routine may be programmed in a suitable programming language.

20 The above embodiment is to be considered in all aspects as illustrative and not restrictive. There may be many modifications, changes, and alterations without departing from the scope or spirit of the main characteristics of the present

invention. All changes within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.